II. Amendments to the Claims

Please replace the claims as previously submitted with the following claims:

1. (Currently Amended) A method for creating application software comprising the steps of:

storing a first set of object models representing data of an application, each object model corresponding to a set of elements, wherein the elements are selected from a finite set of basic object types;

selecting a subset of service objects from a stored set of service objects, wherein each of the service objects includes functionality to parse at least one object model to determine elements and basic object types in the at least one object model and functionality to perform ~~performing~~ a function with respect to elements of each of the basic object types; and

defining a flow process representing an order for operation of the selected subset of service objects and data of the application.

2. (Original) The method of claim 1, wherein the basic object types include:

a primitive,

a class, and

an object array.

3. (Original) The method according to claim 2, wherein the class object type includes a plurality of attributes, each of which is a basic object type.

4. (Original) The method according to claim 2, wherein the object array object type includes a plurality of elements, each of the plurality of elements being of a single basic object type.

5. (Original) The method according to claim 1, wherein at least one of the service objects provides functions with respect to an identified device driver for a resource.

6. (Original) The method according to claim 1, further comprising the step of creating and storing a set of service objects.

7. (Original) The method according to claim 6, wherein the creating and storing step includes creating at least one service object which performs different functions depending upon a basic data type of at least one of the object models.

8. (Previously Amended) A system for creating application software comprising:

means for storing a first set of object models representing data of an application, each object model corresponding to a set of elements, wherein the elements are selected from a finite set of basic object types;

a stored set of service objects each of which includes functionality to parse at least one object model to determine elements and basic object types in the at least one object model and functionality to ~~can~~ perform a function with respect to elements of each of the basic object types; and

means for selecting a subset of the stored set of service objects; and

means for defining a flow process representing an order for operation of the subset of service objects and data of the application.

9. (Original) The system according to claim 8, wherein the means for storing includes:

means for receiving an application model representing data in the application software;

means for classifying each data element in the application model as an object model; and

means for storing each object model from the classifying means.

10. (Previously Amended) The system according to claim 8, wherein at least one of the service models can provide an interface function with at least one resource.

11. (Currently Amended) A system for executing an application program comprising:

a stored set of object models representing data of the application, each object model corresponding to a set of basic object types;

a stored set of service objects, wherein each of the service objects ~~which can~~ performs a function with respect to data of each of the basic object types, wherein ~~the~~ a function of a service object is based upon the set of basic object types of an ~~corresponding~~ object model for data to be acted upon;

means for determining the set of basic object types of an ~~corresponding~~ object model for data upon execution of each of the stored service objects; and

means for executing a function of each service object based upon the determined set of basic object types of the ~~a corresponding~~ object model.